WEST

Generate Collection

Search Results - Record(s) 1 through 10 of 11 returned.

1. Document ID: US 6108635 A

L3: Entry 1 of 11 File: USPT Aug 22, 2000

US-PAT-NO: 6108635
DOCUMENT-IDENTIFIER: US 6108635 A
TITLE: Integrated disease information system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Image

2. Document ID: US 6058387 A

L3: Entry 2 of 11 File: USPT May 2, 2000

US-PAT-NO: 6058387
DOCUMENT-IDENTIFIER: US 6058387 A
TITLE: Dynamic information architecture system and method

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Image

3. Document ID: US 5987454 A

L3: Entry 3 of 11 File: USPT Nov 16, 1999

US-PAT-NO: 5987454
DOCUMENT-IDENTIFIER: US 5987454 A
TITLE: Method and apparatus for selectively augmenting retrieved text, numbers, maps, charts, still pictures and/or graphics, moving pictures and/or graphics and audio information from a network resource

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Image

4. Document ID: US 5862223 A

L3: Entry 4 of 11 File: USPT Jan 19, 1999

US-PAT-NO: 5862223
DOCUMENT-IDENTIFIER: US 5862223 A
TITLE: Method and apparatus for a cryptographically-assisted commercial network system designed to facilitate and support expert-based commerce

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. Desc | Image

WEST

End of Result Set

Generate Collection

L3: Entry 11 of 11 File: USPT Mar 20, 1990

DOCUMENT-IDENTIFIER: US 4910669 A
TITLE: Binary tree multiprocessor

APD:
19870403

BSPR:
Signal pattern matching is to be distinguished from data, or value, pattern matching in which values of each of successive data segments of a sequence of input segments is compared, usually subtractively, against similar reference segments to identify all locations in the database of each input segment. Data pattern matching is often applied in information retrieval systems. One example of such data pattern matching occurs in rule-based systems, where values in "if" clauses of rules are matched against values of working memory, such as that in the paper "DADO: A Parallel Processor for Expert Systems" by S. J. Stolfo et al., "Proceedings of the 1984 International Conference on Parallel Processing," pages 74-82.

BSPR:
In the "Sunday Star-Ledger" newspaper published in Newark, NJ, there appeared on Oct. 5, 1986, an article "Princeton `Brain trust` Top scientists plug into supercomputer" by K. MacPherson. That article mentioned a Cyber 250 single computer capable of about 800 million floating point operations per second (MFLOPS). The article also speaks of a planned ETA-10 computer system being manufactured and using up to eight processors, with liquid nitrogen cooling, costing about $10 million which is said to be capable of 10,000 million floating point operations per second (MFLOPS), i.e. a speed about half of the above-noted computational requirement goal-the units of instructions per second and operations per second being different words for the same thing.

BSPR:
In the data pattern matching arena, one of the faster multiprocessing arrays is the binary tree. Such a tree and its capabilities are discussed at length in "The Tree Machine: A Highly Concurrent Computing Environment" by S. A. Browning in Technical Report (Ph.D. Thesis), Computer Science, California Institute of Technology, 1980. The processing element processor is considered at pages 132-134 and is described as including four main parts: a program store, a bank of data storage registers, an arithmetic logic unit (ALU), and some communication handlers. Current work is represented by, for example, the DADO multiprocessing system outlined in the aforementioned S. Stolfo et al. paper. Communication between nodes in Stolfo et al is by way of a three-link path including an input/output (I/O) link extending either up or down in the tree, a so-called handshake line extending both up and down in the tree, and a third link (upward from the node processor and downward from the node data memory). The handshake lines comprise an unbroken wire network extending throughout the tree, but the manner of preventing internode interference through that network is not shown. A DADO 1023-processing-element system was to have an unpipelined microprocessor at each element and was expected to be able to realize a top processing speed of about 570 million such instructions per second (MIPS).

DEPR:
FIG. 1 depicts a binary tree multiprocessing array in functional diagram form. A host computer 10 receives input signals and provides output data signals for the array in whatever system application makes use of the parallel signal

processing function of the array. The host performs the system interface function for the array and is further coupled to processing elements of the array through a root processing element (PE) 1. Clocking signals are provided from the host to the array for keeping the host and array processing elements operating at compatible rates. Circuits for distributing the clocking signals are not specifically shown but advantageously comprise, for example, a clock signal bus between the host and all processing elements and extending through the backplane of equipment frames including plug-in circuit boards containing the actual processing element circuits.

DEPR:
Each PE includes, with particular reference to PE 1, an external communication processor function 1.2 that handles communication with the host computer 10 (a parent PE in the case of an internal of a leaf PE) by way of a communication line 1.1. That communication line, and other such lines to be hereinafter mentioned between PEs, each is a bidirectional line including an input line part and an output line part. Similarly, the external communication processor function 1.2 communicates with the child PEs 2 and 3 by way of lines 1.3 and 1.4, respectively, and the parent communication lines 2.1 and 3.1 of those two PEs, respectively. Also, a line 1.5 extends from processor function 1.2 to a local message handler processing function 1.6. A line 1.7 extends further to a memory 1.8, and a line 1.9 extends to an execution processor function 1.10. The latter has communication lines 1.11 to the memory 1.8 and 1.13 to a pipelined signal processor function 1.12. The latter function 1.12 further communicates by way of a line 1.14 with the memory 1.8.

DEPR:
By way of background, the concept of using a binary tree machine for parallel matching and the associated broadcast, and resolve types of functions, were taught in a technical report "A Tree Machine for Searching Problems" by J. L. Bentley and H. T. Kung, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pa., Aug. 30, 1979. These functions were implemented in the DADO tree machine mentioned, but not explained, by Stolfo et al. in their aforementioned 1984 paper. A BROADCAST instruction transmits data from the host to all PEs. Such data may comprise, for example, reference data templates against which input data sequences are to be compared, instructions to cause certain programs to be executed (positively or conditionally) in each PE receiving the instruction, and input data signal sequences to be compared against previously stored reference data. A RESOLVE instruction causes the tree to select, e.g., the minimum of a set of values, such as the results of a sliced procedure that reside in respective different PEs. A REPORT instruction typically follows a RESOLVE instruction and sends to the host a selected value from the PE that contained the minimum value.

DEPR:
Host computer 10 initiates a sliced procedure by broadcasting a message SLP, in the form of a pointer, to the tree by way of root PE 1. That message includes one field component, again broadcast in a get/send step pair before execution can begin:

DETL:
Host computer 10 AT&T Personal Computer 6300 Processor functions 1.2, WE.sup.(R) DSP32 Digital Signal 1.6, 1.10, and 1.12; and Processor cooperating with at least a part of memory a communication controller function 1.8

CLPV:
means for interconnecting each of said elements in one of said levels for bidirectionally communicating with only one of said elements, a parent element, in an adjacent level in the direction of said root element, and for bidirectionally communicating with no more than two of said elements, child elements, in an adjacent level in the direction of said leaf level.

ORPL:
"The Tree Machine: A Highly Concurrent Computing Environment", by S. A. Browning, Technical Report, Jan. 1980, Computer Science California Institute of Technology, Pasadena, California 91125, spons Defense Advanced Research Projects Agency.

ORPL:
DADO: A Parallel Processor for Expert Systems, by S. J. Stolfo et al, "Proceedings of the 1984 International Conference on Parallel Processing" pp. 74-82, Department of Computer Science Columbia University, New York City, N.Y. 10027, by S. J. Stolfo.

L3: Entry 9 of 11 File: USPT Nov 2, 1993

DOCUMENT-IDENTIFIER: US 5259066 A
TITLE: Associative program control

APD:
19900416

ABPL:
Methods and an arrangement providing improved expert system performance. A mathematically based method is used in building a rule base that guarantees a complete and consistent rule set, providing an aid for identifying missing rules. The method is used in recognizing input variable patterns in parallel to provide rapid inferencing. An arrangement based on the method is given for a content addressable memory capable of returning an address of an executable routine for each rule of a rule set given the system state variables. The method is employed to provide associative program control, in which program control flow adapts to changing system state variables. A method is described for associative database management, using the new content addressable memory arrangement.

BSPR:
Expert systems are computer programs that provide the skill of an expert in a defined area of expertise. Rule based expert systems capture the skill of an expert by expressing the expertise as a series of rules. The rules are expressed as if-then statements involving symbolic variables. For example, "If conditions (A and B and C) or (B and D and not E) are true, then F is true." makes a logical inference about symbolic variable F based on the program's knowledge of the states of variables A through E. The variables take on the values of true or false and Boolean algebraic rules of logic are used to evaluate the "if" part of the statement. The "then" part of the statement performs an action based on the outcome of the evaluation of the "if" part. In this example, F is set "true" if either set of conditions in parentheses evaluates to "true". Otherwise F is set "false".

BSPR:
Another problem faced by expert system designers is the relatively slow speed with which inferences can be drawn. Even though computer speeds have increased dramatically, as more knowledge is built into expert systems, the number of potential decision paths increases exponentially and overpowers the computer improvement. This is known as the combinatorial explosion in which the number of possible outcomes for N variables is equal to 2 raised to the Nth power. The serial nature of most computers and the lack of a viable method to process inferences in a parallel manner efficiently have hampered obtaining a satisfactory solution to this problem.

BSPR:
The problem is akin to maintaining a database for rapid retrieval of information. A directory of keywords can be used to point to the location of randomly stored data. As keywords are added to and deleted from the directory over the lifespan of the database, they have to be inserted in a sorted order, to enable more rapid access. This entails time consuming computational effort. The retrieval then requires a search of the directory to find the keyword which is also time consuming. To reduce the time, hashing methods have been employed. However, hashing takes time and runs into difficulty as the stored data nears the capacity of the storage area.

BSPR:
In keeping with this object and with still others which will become apparent

as the description proceeds, the important characteristics of the invention are: a mathematically based method of building a rule base, a method and arrangements for parallel association of variables, a method of providing associative program control, and a method for providing associative database management.

BSPR:
A preferred arrangement for providing associative database management comprises storing key words in content addressable memory which delivers address pointers to locations in standard memory where the bulk of the related data is stored. A preferred method eliminates the need to provide for multiple entries of key words in the content addressable memory by examining the content addressable memory for the word prior to storing it. If the word already exists in the memory, it is not stored again, but the related stored data is modified, if necessary, to reflect the new information.

DRPR:
FIG. 10 shows a preferred computer architecture with associative program control and associative data base management.

DEPR:
FIG. 7 illustrates how this methodology could be implemented with standard hardware available in today's computers. Two bits would have to be stored for each variable in each rule in order to represent the three possible variable states. A preferred method would encode the rule information in one computer word as shown on row 71 and encode validity information (mask) in another computer word as shown on row 72. The four rules of Table 62 are represented in the same order on rows 71, 73, 74, 75 with their associated masks below them. The rule 1's and 0's appear unchanged, and the e's are replaced by 0's. The validity bit is 1 for 1 or 0 in Table 62; the validity bit is 0 for each e.

DEPR:
CPU 101 accesses data stored in RAM 107 via data bus 105. CAM 103 doesn't have to handle multiple occurrences of the same stored word because of the mutually exclusive rule construction of the new methodology. This allows using a simpler circuit within the CAM. The same simpler CAM circuitry can be used for CAM 106 to store key words to return pointers to related data in RAM 107, if duplicate key words are not stored. This can be assured by first presenting each new key word to CAM 106 to check if it matches any word already in the CAM. The new word would be stored only if no match is reported by CAM 106. If a match is reported, the new word would not be stored. However, if the application required the related data in RAM 107 to be modified, this would be done. No sorting, hashing or other scheme is required to store the keywords into CAM 106. No sequential search is required to retrieve a keyword. Thus database management is simplified and processing time reduced.

DEPR:
The new architecture of FIG. 10 is seen to be generally applicable to all processing. The ability of CAM's 103 and 106 to recognize patterns can greatly speed up applications needing that capability. Fast retrieval of objects based on matching attributes can be readily implemented. Database management has already been discussed as benefiting from the parallel search capability of a CAM. Interpretive languages can use the parallel search capability to reduce the time to match key word inputs. Intelligent control will benefit from the fast response time, large rule base capacity, and the well defined responses of rule based systems.

DEPR:
Data bit line 123a brings the input signal to a bidirectional buffer 122 whose direction of transmission is controlled by the logic level of R/W line 125 generated by conventional decoding (not shown) of control bits. In the three modes for storing data, signals on line 123a pass through buffer 122 to line 123b. Line 123b brings the signal to each rule bit storage cell represented by the area between lines 120 and 121. When storing the rule bit, the logic level is set to a 1 on line 126 of the rule selected by conventional decoding (not shown) of control and address bits (111 and 112 in FIG. 11). Similarly, line 127 is set to a 1 when storing a valid bit and line 128 is set to a 1 when storing an action address bit.

DEPR:
When in the action address readout mode, the logic level is set to a 1 on line 1214 generated by conventional decoding (not shown) of control bits. Line 1214 is otherwise set to 0. When line 1214 is set to 1, the level stored in storage element 1219 passes through AND gate 1220 to line 124b (referred to as an enable signal on line 114 of FIG. 11). If latch reset line 1212 is activated prior to the rule matching mode, then only one element 1219 will contain a 1 to force its line 124b to a 1 level. The line 124b that is at a 1 level will enable the logic level stored in element 1213 of each rule bit storage cell to be passed to its corresponding data line 123b via its AND gate 129. The level on line 123b passes through bidirectional buffer 122, controlled by R/W line 125, to data bit line 123a. The action address is thus placed on the data lines (113 of FIG. 11). Reset line 1212 can be activated at the end of the readout mode. Reset line 1212 is not required if element 1219 is a D-type flip-flop.

CLPR:
16. A rule based expert system as defined in claim 14 wherein said CAM reports a reporting word associated with said matching rule in place of said matching rule and said CAM further includes:

CLPR:
19. A rule based expert system as defined in claim 17 wherein said PCAM reports a reporting word associated with said matching rule in place of said matching rule and said PCAM further includes:

CLPR:
22. A rule based expert system as defined in claim 20 wherein said MCAM reports a reporting word associated with said matching rule in place of said matching rule and said MCAM further includes:

ORPL:
Chisvin et al., "Content-Addressable and Associative Memory: Alternatives to the Ubiquitous RAM", IEEE Computer, Jul. 1989, 51-64.

ORPL:
Ogura et al., "A 20 Kb CMOS Associative Memory LSI for Artificial Intelligence Machines", Proc. IEEE Intl. Conf. Computer Design: VLSI in Computers, 1986, 574-577.

WEST

Generate Collection

L3: Entry 9 of 11 File: USPT Nov 2, 1993

DOCUMENT-IDENTIFIER: US 5259066 A
TITLE: Associative program control

APD:
19900416

ABPL:
Methods and an arrangement providing improved expert system performance. A mathematically based method is used in building a rule base that guarantees a complete and consistent rule set, providing an aid for identifying missing rules. The method is used in recognizing input variable patterns in parallel to provide rapid inferencing. An arrangement based on the method is given for a content addressable memory capable of returning an address of an executable routine for each rule of a rule set given the system state variables. The method is employed to provide associative program control, in which program control flow adapts to changing system state variables. A method is described for associative database management, using the new content addressable memory arrangement.

BSPR:
Expert systems are computer programs that provide the skill of an expert in a defined area of expertise. Rule based expert systems capture the skill of an expert by expressing the expertise as a series of rules. The rules are expressed as if-then statements involving symbolic variables. For example, "If conditions (A and B and C) or (B and D and not E) are true, then F is true." makes a logical inference about symbolic variable F based on the program's knowledge of the states of variables A through E. The variables take on the values of true or false and Boolean algebraic rules of logic are used to evaluate the "if" part of the statement. The "then" part of the statement performs an action based on the outcome of the evaluation of the "if" part. In this example, F is set "true" if either set of conditions in parentheses evaluates to "true". Otherwise F is set "false".

BSPR:
Another problem faced by expert system designers is the relatively slow speed with which inferences can be drawn. Even though computer speeds have increased dramatically, as more knowledge is built into expert systems, the number of potential decision paths increases exponentially and overpowers the computer improvement. This is known as the combinatorial explosion in which the number of possible outcomes for N variables is equal to 2 raised to the Nth power. The serial nature of most computers and the lack of a viable method to process inferences in a parallel manner efficiently have hampered obtaining a satisfactory solution to this problem.

BSPR:
The problem is akin to maintaining a database for rapid retrieval of information. A directory of keywords can be used to point to the location of randomly stored data. As keywords are added to and deleted from the directory over the lifespan of the database, they have to be inserted in a sorted order, to enable more rapid access. This entails time consuming computational effort. The retrieval then requires a search of the directory to find the keyword which is also time consuming. To reduce the time, hashing methods have been employed. However, hashing takes time and runs into difficulty as the stored data nears the capacity of the storage area.

BSPR:
In keeping with this object and with still others which will become apparent

as the description proceeds, the important characteristics of the invention are: a mathematically based method of building a rule base, a method and arrangements for parallel association of variables, a method of providing associative program control, and a method for providing associative database management.

BSPR:
A preferred arrangement for providing associative database management comprises storing key words in content addressable memory which delivers address pointers to locations in standard memory where the bulk of the related data is stored. A preferred method eliminates the need to provide for multiple entries of key words in the content addressable memory by examining the content addressable memory for the word prior to storing it. If the word already exists in the memory, it is not stored again, but the related stored data is modified, if necessary, to reflect the new information.

DRPR:
FIG. 10 shows a preferred computer architecture with associative program control and associative data base management.

DEPR:
FIG. 7 illustrates how this methodology could be implemented with standard hardware available in today's computers. Two bits would have to be stored for each variable in each rule in order to represent the three possible variable states. A preferred method would encode the rule information in one computer word as shown on row 71 and encode validity information (mask) in another computer word as shown on row 72. The four rules of Table 62 are represented in the same order on rows 71, 73, 74, 75 with their associated masks below them. The rule 1's and 0's appear unchanged, and the e's are replaced by 0's. The validity bit is 1 for 1 or 0 in Table 62; the validity bit is 0 for each e.

DEPR:
CPU 101 accesses data stored in RAM 107 via data bus 105. CAM 103 doesn't have to handle multiple occurrences of the same stored word because of the mutually exclusive rule construction of the new methodology. This allows using a simpler circuit within the CAM. The same simpler CAM circuitry can be used for CAM 106 to store key words to return pointers to related data in RAM 107, if duplicate key words are not stored. This can be assured by first presenting each new key word to CAM 106 to check if it matches any word already in the CAM. The new word would be stored only if no match is reported by CAM 106. If a match is reported, the new word would not be stored. However, if the application required the related data in RAM 107 to be modified, this would be done. No sorting, hashing or other scheme is required to store the keywords into CAM 106. No sequential search is required to retrieve a keyword. Thus database management is simplified and processing time reduced.

DEPR:
The new architecture of FIG. 10 is seen to be generally applicable to all processing. The ability of CAM's 103 and 106 to recognize patterns can greatly speed up applications needing that capability. Fast retrieval of objects based on matching attributes can be readily implemented. Database management has already been discussed as benefiting from the parallel search capability of a CAM. Interpretive languages can use the parallel search capability to reduce the time to match key word inputs. Intelligent control will benefit from the fast response time, large rule base capacity, and the well defined responses of rule based systems.

DEPR:
Data bit line 123a brings the input signal to a bidirectional buffer 122 whose direction of transmission is controlled by the logic level of R/W line 125 generated by conventional decoding (not shown) of control bits. In the three modes for storing data, signals on line 123a pass through buffer 122 to line 123b. Line 123b brings the signal to each rule bit storage cell represented by the area between lines 120 and 121. When storing the rule bit, the logic level is set to a 1 on line 126 of the rule selected by conventional decoding (not shown) of control and address bits (111 and 112 in FIG. 11). Similarly, line 127 is set to a 1 when storing a valid bit and line 128 is set to a 1 when storing an action address bit.

DEPR:
When in the action address readout mode, the logic level is set to a 1 on line 1214 generated by conventional decoding (not shown) of control bits. Line 1214 is otherwise set to 0. When line 1214 is set to 1, the level stored in storage element 1219 passes through AND gate 1220 to line 124b (referred to as an enable signal on line 114 of FIG. 11). If latch reset line 1212 is activated prior to the rule matching mode, then only one element 1219 will contain a 1 to force its line 124b to a 1 level. The line 124b that is at a 1 level will enable the logic level stored in element 1213 of each rule bit storage cell to be passed to its corresponding data line 123b via its AND gate 129. The level on line 123b passes through bidirectional buffer 122, controlled by R/W line 125, to data bit line 123a. The action address is thus placed on the data lines (113 of FIG. 11). Reset line 1212 can be activated at the end of the readout mode. Reset line 1212 is not required if element 1219 is a D-type flip-flop.

CLPR:
16. A rule based expert system as defined in claim 14 wherein said CAM reports a reporting word associated with said matching rule in place of said matching rule and said CAM further includes:

CLPR:
19. A rule based expert system as defined in claim 17 wherein said PCAM reports a reporting word associated with said matching rule in place of said matching rule and said PCAM further includes:

CLPR:
22. A rule based expert system as defined in claim 20 wherein said MCAM reports a reporting word associated with said matching rule in place of said matching rule and said MCAM further includes:

ORPL:
Chisvin et al., "Content-Addressable and Associative Memory: Alternatives to the Ubiquitous RAM", IEEE Computer, Jul. 1989, 51-64.

ORPL:
Ogura et al., "A 20 Kb CMOS Associative Memory LSI for Artificial Intelligence Machines", Proc. IEEE Intl. Conf. Computer Design: VLSI in Computers, 1986, 574-577.